ABSTRACT OF THE DISCLOSURE

A computer software system includes interdependent collections of software components. That is, at the architectural level, software components and their interdependencies are two distinct equally important entities. The software components represent the core functional pieces of an application and deal with concepts specific to an application domain. Interdependencies relate to concepts orthogonal to the problem domain in most applications, such as transportation, sharing of resources and synchronization constraints among components. An architectural description language which represents activities and dependencies between activities as separate entities. Dependencies are managed by coordination processes associated with the dependency. Activities and dependencies are connected through ports which encode interfaces between activities and coordination processes. The language may also represent resources which may be understood as the output of some activity beyond the scope of the system. Each entity, i.e., activity, dependency, port or resource, may also have attributes which are name value pairs, specifying additional information about the entity. Attributes may be inherited. That is, activities and dependencies may be specialized into particular versions of an activity or dependency. The attributes for a particular activity are inherited by its specializations. An editor repository and design assistant may be based on this language to provide a system that automatically generates executable code.